NATIONAL INSTITUTE OF TECHNOLOGY WARANGAL

SCHEME OF INSTRUCTION AND SYLLABI
FOR M.TECH PROGRAM

Effective from 2014-15

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY WARANGAL

VISION

Towards a Global Knowledge Hub, striving continuously in pursuit of excellence in Education, Research, Entrepreneurship and Technological services to the society

MISSION

- Imparting total quality education to develop innovative, entrepreneurial and ethical future professionals fit for globally competitive environment.

- Allowing stake holders to share our reservoir of experience in education and knowledge for mutual enrichment in the field of technical education.

- Fostering product oriented research for establishing a self-sustaining and wealth creating centre to serve the societal needs.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION

Attaining global recognition in Computer Science & Engineering education, research and training to meet the growing needs of the industry and society.

MISSION

- Imparting quality education through well-designed curriculum in tune with the challenging software needs of the industry.

- Providing state-of-art research facilities to generate knowledge and develop technologies in the thrust areas of computer science and engineering.

- Developing linkages with world class organizations to strengthen industry-academia relationships for mutual benefit.

GRADUATE ATTRIBUTES

The Graduate Attributes are the knowledge skills and attitudes which the students have at the time of graduation. These attributes are generic and are common to all engineering programs. These Graduate Attributes are identified by National Board of Accreditation.

1. Scholarship of Knowledge: Acquire in-depth knowledge of specific discipline or professional area, including wider and global perspective, with an ability to discriminate, evaluate, analyze and synthesize existing and new knowledge, and integration of the same for enhancement of knowledge.
2. Critical Thinking: Analyze complex engineering problems critically, apply independent judgment for synthesizing information to make intellectual and/or creative advances for conducting research in a wider theoretical, practical and policy context.
3. Problem Solving: Think laterally and originally, conceptualize and solve engineering problems, evaluate a wide range of potential solutions for those problems and arrive at feasible, optimal solutions after considering public health and safety, cultural, societal and environmental factors in the core areas of expertise.
4. Research Skill: Extract information pertinent to unfamiliar problems through literature survey and experiments, apply appropriate research methodologies, techniques and tools, design, conduct experiments, analyze and interpret data, demonstrate higher order skill and view things in a broader perspective, contribute individually/in group(s) to the development of scientific/technological knowledge in one or more domains of engineering.
5. Usage of modern tools: Create, select, learn and apply appropriate techniques, resources, and modern engineering and IT tools, including prediction and modeling, to complex engineering activities with an understanding of the limitations.
6. Collaborative and Multidisciplinary work: Possess knowledge and understanding of group dynamics, recognize opportunities and contribute positively to collaborative-multidisciplinary scientific research, demonstrate a capacity for self-management and teamwork, decision-making based on open-mindedness, objectivity and rational analysis in order to achieve common goals and further the learning of themselves as well as others.
7. Project Management and Finance: Demonstrate knowledge and understanding of engineering and management principles and apply the same to one's own work, as a member and leader in a team, manage projects efficiently in respective disciplines and multidisciplinary environments after consideration of economical and financial factors.
8. Communication: Communicate with the engineering community, and with society at large, regarding complex engineering activities confidently and effectively, such as, being able to comprehend and write effective reports and design documentation by adhering to

appropriate standards, make effective presentations, and give and receive clear instructions.

9. Life-long Learning: Recognize the need for, and have the preparation and ability to engage in life-long learning independently, with a high level of enthusiasm and commitment to improve knowledge and competence continuously.
10. Ethical Practices and Social Responsibility: Acquire professional and intellectual integrity, professional code of conduct, ethics of research and scholarship, consideration of the impact of research outcomes on professional practices and an understanding of responsibility to contribute to the community for sustainable development of society.
11. Independent and Reflective Learning: Observe and examine critically the outcomes of one’s actions and make corrective measures subsequently, and learn from mistakes without depending on external feedback.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

M.TECH IN COMPUTER SCIENCE AND ENGINEERING

PROGRAM EDUCATIONAL OBJECTIVES

PEO1.	Design, develop and test software systems for engineering applications.
PEO2.	Analyze technical solutions to computational problems and develop efficient algorithms.
PEO3.	Work in multi-disciplinary teams to specify software requirements and to achieve project goals.
PEO4.	Communicate effectively and demonstrate professional ethics with societal responsibilities.
PEO5.	Engage in lifelong learning to keep pace with changing landscape of technologies for professional advancement.

Mapping of Mission statements with program educational objectives

Mission Statement	PEO1	PEO2	PEO3	PEO4	PEO5
MS1	3	2	1	1	2
MS2	2	2	2		2
MS3	2	2	2	2	1

Mapping of program educational objectives with graduate attributes

	GA1	GA2	GA3	GA4	GA5	GA6	GA7	GA8	GA9	GA10	GA11
PEO1	2	1	3		2	1		2	1	1	
PEO2	3	3	2	2				2			1
PEO3	3	2		2	2	2	2	2			1
PEO4						2	1	2		3	
PEO5	2	2		2	2		2	2	2	1	2

PROGRAM OUTCOMES: At the end of the program the student will be able to:

PO1	Apply concepts of theoretical computer science to design software systems.
PO2	Design and develop processes to meet targeted needs within realistic economic, social, safety and security constraints.
PO3	Analyze and prevent the impact of threats, risks, and vulnerabilities in software systems.
PO4	Apply techniques and tools to build robust, reliable and maintainable software.
PO5	Model efficient scalable software systems.
PO6	Communicate effectively and exhibit leadership skills.
PO7	Work in teams with professional ethics to achieve project goals.
PO8	Engage in life-long learning for professional advancement.

Mapping of program outcomes with program educational objectives

	PEO1	PEO2	PEO3	PEO4	PEO5
PO1	3	3			1
PO2	2	2	1		1
PO3	3	2	1	2	2
PO4	2	2	2		1
PO5	3	3	2		
PO6	1	1	3	3	3
PO7			2	3	2
PO8			1	3	3

CURRICULAR COMPONENTS

Degree Requirements for M. Tech. in Computer Science and Engineering

Category of Courses	Credits Offered	Min. credits to be earned
Program Core Courses (PCC)	≥ 38	38
Departmental Elective Courses (DEC)	≥ 15	15
Program major Project (PCC)	26	26
Total	**≥ 79**	**79**

SCHEME OF INSTRUCTION

M.Tech. (Computer Science and Engineering) Course Structure

M. Tech. I - Year I - Semester

S. No.	Course Code	Course Title	L	T	P	Credits	Cat. Code
1	CS5101	Advanced Algorithms and Complexity	4	0	0	4	PCC
2	CS5102	Advanced Operating Systems	4	0	0	4	PCC
3	CS5103	Business Intelligence	4	0	0	4	PCC
4	CS5104	Advanced Operating Systems Lab	0	0	3	2	PCC
5	CS5105	Business Intelligence Lab	0	0	3	2	PCC
6		Elective - 1	3	0	0	3	DEC
7		Elective - 2	3	0	0	3	DEC
8		Elective - 3	3	0	0	3	DEC
		TOTAL	**21**	**0**	**6**	**25**	

M.Tech. I - Year II - Semester

S. No.	Course Code	Course Title	L	T	P	Credits	Cat. Code
1	CS5151	High Performance Computing	4	0	0	4	PCC
2	CS5152	Socket Programming and Network Security	4	0	0	4	PCC
3	CS5153	Advanced Data Mining	4	0	0	4	PCC
4	CS5154	High Performance Computing Lab	0	0	3	2	PCC
5	CS5155	Socket Programming and Network Security Lab	0	0	3	2	PCC
6		Elective – 4	3	0	0	3	DEC
7		Elective – 5	3	0	0	3	DEC
8	CS5191	Seminar	0	0	3	2	PCC
		TOTAL	**18**	**0**	**9**	**24**	

II - Year I - Semester

S. No.	Course Code	Course Title	L	T	P	Credits	Cat. Code
1	CS6142	Comprehensive Viva	0	0	0	4	PCC
2	CS6149	Dissertation Work – Part A	0	0	0	8	PCC
		TOTAL	**0**	**0**	**0**	**12**	

II - Year II - Semester

S. No.	Course Code	Course Title	L	T	P	Credits	Cat. Code
1	CS6199	Dissertation Work – Part B	0	0	0	18	PCC
		TOTAL	**0**	**0**	**0**	**18**	

List of Electives

I Year I Semester

- CS5111 Machine Learning
- CS5112 Human Computer Interaction
- CS5113 Computer Vision
- CS5114 Intelligent Agents
- CS5121 Object Oriented Software Engineering
- CS5122 Wireless and Mobile Networks
- CS5123 Real Time Systems
- CS5124 Model-driven Software Development
- CS5131 Distributed Computing
- CS5132 Advanced Databases
- CS5133 Game Theory
- CS5134 Semantic Web
- CS5135 Big Data Analytics

I Year II Semester

- CS5161 Security and Privacy
- CS5162 Intrusion Detection Systems
- CS5163 Social Media Analysis
- CS5164 Randomized and Approximation Algorithms
- CS5171 Pattern Recognition
- CS5172 Cloud Computing
- CS5173 Bio-Informatics
- CS5174 Design Patterns

DETAILED SYLLABUS

CS5101	Advanced Algorithms and Complexity	PCC	4 – 0 – 0	4 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Analyze worst-case running times of algorithms using asymptotic analysis.
CO2	Prove the correctness of algorithms using inductive proofs and invariants.
CO3	Analyze randomized algorithms with respect to expected running time, probability of error using tail inequalities
CO4	Classify problems into different complexity classes corresponding to both deterministic and randomized algorithms
CO5	Analyze approximation algorithms including algorithms that are PTAS and FPTAS.

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	2	2		1	2			
CO2	2		2		2			
CO3	1	2	3	2	2			
CO4		2			2			
CO5	1	2	2	2	1		1	1

Detailed syllabus

Algorithm design techniques – Dynamic programming: Matrix chain multiplication, Optimal BST, Greedy algorithms – Shortest path algorithm, MST, Amortized analysis, Data structures for disjoint sets, Divide-and-Conquer- Karatsuba integer multiplication, Large integer multiplications using FFT, NP-Completeness: Poly-time, Poly-time verification, reducibility, NP-Complete problems, Approximation algorithms, Randomized algorithms: Las Vegas and Monte Carlo, Game-Theoretic Techniques: Game Tree Evaluation, The Minimax Principle, Randomness and Non-uniformity, Moments and Deviations: Occupancy Problems, The Markov and Chebyshev, Inequalities, Randomized Selection, Two-Point Sampling, The Stable Marriage Problem, The Coupon Collector's Problem, Tail Inequalities: The Chernoff Bound, Routing in a Parallel Computer, A Wiring Problem.

Reading:

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, *Introduction to Algorithms*, 2nd Edition, The MIT Press, 2001.
2. C.H. Papadimitriou. *Complexity Theory.* Addison-Wesley, Reading, MA, 1994.
3. Rajeev Motwani and Prabhakar Raghavan, *Randomized Algorithms,* Cambridge University Press, 1995.
4. Garey Michael R, Johnson davis S, *Computers and Intractability: A Guide the theory of NP-Incompleteness,* W.H. Freeman & Co.1979.

CS5102	Advanced Operating Systems	PCC	4 – 0 – 0	4 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Design and implement Unix kernel data structures and algorithms
CO2	Analyze synchronization problems in uniprocessor and multiprocessor systems
CO3	Evaluate the scheduling requirements of different types of processes and find their solutions
CO4	Implement user level thread library and mimic the behaviour of Unix kernel for scheduling, synchronization and signals.

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1		3		2	3		3	
CO2	3	3		3	3			
CO3	3	3		3	3			
CO4	3	3		3	3		2	

Detailed syllabus

Introduction to UNIX : History, Need of change, Standards

The process and the kernel : Mode, space and context, Process abstraction, executing in kernel mode, synchronization by blocking interrupts, process scheduling, signals, process creation, termination, awaiting process termination, zombie processes

Introduction to Threads : Fundamental abstractions, Lightweight process design, issues to consider, User level thread libraries, scheduler activations, Multi threading on Solaris, Pthreads library, Thread library implementation Using ucontext_t structures

Signals and Session Management : Signal generation and handling, Unreliable signals, Reliable signals, Signals in SVR4, Signals implementation, Exceptions, Process Groups and Terminal management, SVR4 Sessions architecture

Process Scheduling : Clock interrupt handling, Scheduler Goals, Traditional UNIX scheduling

Scheduling case studies

Synchronization and Multiprocessing : Introduction, Synchronization in Traditional UNIX Kernels, Multiprocessor Systems, Multiprocessor synchronization issues, Semaphores, spin locks, condition variables Read-write locks for multiprocessor systems, Reference counts and other considerations.

File system interface and framework : The user interface to files, File systems, Special files, File system framework, The Vnode/Vfs architecture, Implementation Overview, File System dependent objects, Mounting a file system, Operations on files.

File System Implementations : System V file system (s5fs) implementation, Berkeley FFS, FFS functionality enhancements and analysis, Temporary file systems, Buffer cache and other special-purpose file systems

Distributed File Systems : Network File System (NFS), Remote File Sharing (RFS)

Advanced File Systems : Limitations of traditional file systems, Sun-FFS, Journaling approach 4.4 BSD, Log-Structured file system, Meta logging Episode FS, Watchdogs, 4.4 BSD portal FS, Stackable FS layers, 4.4 BSD FS interface Kernel Memory Allocators: Resource map allocator, Simple power-of-two allocator, McCusick-Karels Allocator, Buddy system, SVR4 Lazy Buddy allocator, OSF/1 Zone Allocator, Hierarchical Allocator, Solaris Slab Allocator

Reading:

1. Uresh Vahalia, UNIX Internals, Pearson Education, 2005.
2. Richard Stevens, Stephen Rago, Advanced Programming in the UNIX Environment, Pearson Education, 2/e.

CS5103	Business Intelligence	PCC	4 – 0 – 0	4 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Indentify the need for data warehouse for large organizations.
CO2	Determine the data sources to populate data warehouse.
CO3	Design data warehouse models using appropriate schemas.
CO4	Develop data warehouse for a domain using Data warehouse tools.
CO5	Design data warehouse to meet business objectives.
CO6	Apply data analysis techniques for building Decision support system.

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1							1	1
CO2		2						
CO3	2	2		2	2			1
CO4	2	2		2	2		2	
CO5	2	2		2	2		2	
CO6	1		2	3				1

Detailed syllabus

Business Intelligence Introduction – Definition, Leveraging Data and Knowledge for BI, BI Components, BI Dimensions, Information Hierarchy, Business Intelligence and Business Analytics. BI Life Cycle. Data for BI - Data Issues and Data Quality for BI.

BI Implementation - Key Drivers, Key Performance Indicators and operational metrics, BI Architecture/Framework, Best Practices, Business Decision Making.

Business Analytics – Objective Curve, Web Analytics and Web Intelligence, Customer Relationship Management.

Business/Corporate Performance Management - Dash Boards and Scorecards, Business Activity Monitoring, Six Sigma.

Advanced BI – Big Data and BI, Social Networks, Mobile BI, emerging trends.

Working with BI Tools – Pentaho etc.

Overview of managerial, strategic and technical issues associated with Business Intelligence and Data Warehouse design, implementation, and utilization. Critical issues in planning, physical design process, deployment and ongoing maintenance.

Data Warehousing (DW):

Data Warehouse (DW) Introduction & Overview; Data Marts

DW architecture – DW components, Implementation options; Meta Data, Information delivery.

ETL - Data Extraction, Data Transformation – Conditioning, Scrubbing, Merging, etc., Data Loading, Data Staging, Data Quality.

Dimensional Modeling - Facts, dimensions, measures, examples; Schema Design – Star and Snowflake, Fact constellation, Slow changing Dimensions.

OLAP - OLAP Vs OLTP, Multi-Dimensional Databases (MDD); OLAP – ROLAP, MOLAP, HOLAP;

Data Warehouse Project Management - Critical issues in planning, physical design process, deployment and ongoing maintenance.

Reading:

1. Efraim Turban, Ramesh Sharda, Jay Aronson, David King, Decision Support and Business Intelligence Systems, 9th Edition, Pearson Education, 2009.
2. David Loshin, Business Intelligence - The Savy Manager's Guide Getting Onboard with Emerging IT, Morgan Kaufmann Publishers, 2009.
3. Paulraj Punniah: Data Warehousing Fundamentals: A comprehensive guide for IT professionals, John Wiley publications, 2001.

CS5104	Advanced Operating Systems Lab	PCC	0 – 0 – 3	2 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Implement basic/UNIX kernel level algorithms.
CO2	Implement the user level thread library and mimic the behaviour of UNIX kernel for scheduling, synchronization and signals.
CO3	Implement File system image in a file and NFS using RPC.

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1		3		2	3		3	
CO2	3	3		3	3		2	
CO3	2	3		3	3			

Detailed syllabus

1. Write Command Interpreter Programs which accepts some basic Unix commands and displays the appropriate result. Each student should write programs for at least six commands.
2. Study the concept of Signals and write a program for Context Switching between two processes using alarm signals.
3. Study pthreads and implement the following: Write a program which shows the performance improvement in using threads as compared with process.(Examples like Matrix Multiplication, Hyper quicksort, Merge sort, Traveling Sales Person problem)
4. Create your own thread library, which has the features of pthread library by using appropriate system calls (UContext related calls). Containing functionality for creation, termination of threads with simple round robin scheduling algorithm and synchronization features.
5. Implement all CPU Scheduling Algorithms using your thread library
6. Study the concept of Synchronization and implement the classical synchronization problems using Semaphores, Message queues and shared memory (minimum of 3 problems)
7. A complete file system implementation inside a disk image file.
8. NFS server and NFS client implementation using RPC

Reading:

1. Richard Stevens, Stephen Rago, Advanced Programming in the UNIX Environment, Pearson Education, 2/e

CS5105	Business Intelligence Lab	PCC	0 – 0 – 3	2 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Build the data cubes with SQL
CO2	Implement data pre-processing techniques on data
CO3	Implement OLAP operations and multi-dimensional modeling
CO4	Generate multi dimensional reports using OLAP COGNOS Tool.
CO5	Develop data warehouse for a domain using open source Data warehouse tools.

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1				2				
CO2		2		2				1
CO3		2		2				2
CO4					1			
CO5	2	2		2	2			1

Detailed syllabus

BI Implementation - Key Drivers, Key Performance Indicators and operational metrics, BI Architecture/Framework, Best Practices, Business Decision Making.

Business Analytics – Objective Curve, Web Analytics and Web Intelligence, Customer Relationship Management.

Dash Boards and Scorecards,

Advanced BI – Big Data and BI, Social Networks, Mobile BI, emerging trends.

Working with BI Tools – Pentaho etc.

DW architecture – DW components, Implementation options; Meta Data, Information delivery.

ETL - Data Extraction, Data Transformation – Conditioning, Scrubbing, Merging, etc., Data Loading, Data Staging, Data Quality.

Dimensional Modeling - Facts, dimensions, measures, examples; Schema Design – Star and Snowflake, Fact constellation, Slow changing Dimensions.

OLAP - OLAP Vs OLTP, Multi-Dimensional Databases (MDD); OLAP – ROLAP, MOLAP, HOLAP;

Reading:

1. Efraim Turban, Ramesh Sharda, Jay Aronson, David King, Decision Support and Business Intelligence Systems, 9th Edition, Pearson Education, 2009.
2. David Loshin, Business Intelligence - The Savy Manager's Guide Getting Onboard with Emerging IT, Morgan Kaufmann Publishers, 2009.
3. Paulraj Punniah: Data Warehousing Fundamentals: A comprehensive guide for IT professionals, John Wiley publications, 2001.

CS5151	High Performance Computing	PCC	4 – 0 – 0	4 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Design and analyze the parallel algorithms for real world problems and implement them on available parallel computer systems.
CO2	Optimize the performance of a parallel program to suit a particular hardware and software environment.
CO3	Write Programs using accelerator technologies of GPGPUs with CUDA, OpenCL.
CO4	Design algorithms suited for Multicore processor systems using OpenCL, OpenMP, Threading techniques.
CO5	Analyze the communication overhead of interconnection networks and modify the algorithms to meet the requirements.

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	1	2	2	1	1		1	
CO2		2	2	1	1		1	
CO3	1	2	1	1	2		1	
CO4	1	2		1	1		1	
CO5	2	2			2			

Detailed syllabus

Introduction: Implicit parallelism, Limitations of memory system performance, control structure, communication model, physical organization, communication costs of parallel platforms, Routing mechanisms for interconnection networks, Mapping techniques

Parallel algorithm design: Preliminaries, decomposition techniques, tasks and interactions, mapping techniques for load balancing, methods for reducing interaction overheads, parallel algorithm models

Basic communication operations: Meaning of all-to-all, all-reduce, scatter, gather, circular shift and splitting routing messages in parts. Analytical modeling of parallel programs: sources of overhead, performance metrics, the effect of granularity on performance, scalability of parallel systems, minimum execution time, minimum cost-optimal execution time, asymptotic analysis of parallel programs

Programming using message passing paradigm: Principles, building blocks, MPI, Topologies and embedding, Overlapping communication and computation, collective communication operations, Groups and communicators

Programming shared address space platforms: Threads, POSIX threads, Synchronization primitives, attributes of threads, mutex and condition variables, Composite synchronization constructs, OpenMP Threading Building blocks; An Overview of Memory Allocators, An overview of Intel Threading building blocks;

An Overview of Brief History of GPUs; An Overview of GPU Programming; An Overview of GPU Memory Hierarchy Features; Introduction to Heterogeneous Computing – OpenCL; The OpenCL Kernel, The OpenCL Memory Model, The OpenCL Execution Model; OpenCL Platform and Devices; OpenCL Execution Environment, An Overview of OpenCL API; Heterogeneous Programming in OpenCL

An Overview of CUDA enabled NVIDIA GPUs, Introduction to CUDA C, Parallel Programming in CUDA C;

Dense Matrix Algorithms: matrix vector multiplication, matrix-matrix multiplication, solving system of linear equations,

Sorting: Sorting networks, Bubble sort, Quick sort, Bucket sort and other sorting algorithms

Graph algorithms: Minimum spanning tree, single source shortest paths, all-pairs shortest paths, Transitive closure, connected components, algorithms for sparse graphs

Reading:

1. Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar : Introduction to Parallel Computing, Second Edition Pearson Education, 2007
2. Benedict R Gaster, Lee Howes, David R Kaeli Perhaad Mistry Dana Schaa, *Heterogeneous Computing* with OpenCL McGraw-Hill, Inc. Newyork , 2011
3. *Michael J. Quinn,* Parallel Programming in C with MPI and OpenMP McGraw-Hill International Editions, Computer Science Series, 2004
4. Jason Sanders, Edward Kandrot, CUDA By Example – An Introduction to General-Purpose GPU Programming, Addison Wesley, 2011.

CS5152	Socket Programming and Network Security	PCC	4 – 0 – 0	4 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Identify and compare network simulation tools
CO2	Design and program client server applications.
CO3	Evaluate the security vulnerabilities in a network.
CO4	Analyze the network security tools and authentication applications.

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	2	2	3					
CO2	1	2	3	3	2	2	1	1
CO3		3	3	2		1	1	2
CO4	2	2	3	3	1	2	1	2

Detailed syllabus

Introduction to ISO's OSI Network Architecture, Internet Model, IP Design and Implementation, Internetworking and routing protocols, Transport layer services and variants, Peer to Peer Networks, Application Layer protocols, Introduction to network Security and associated techniques, Firewall Design principles, VPNs, Worms, Viruses, Vaccine Programs, Security of Network Layer, Security of Application layer protocols, BSD sockets, Elementary and Advanced system calls, Raw sockets: Raw Socket Creation, Raw socket output, raw socket input, packet sniffing and routing algorithms: Router IOS- Static and Default Routing-Interior Gateway Routing Protocols: RIP V1&V2, OSPF, EIGRP- Exterior Gateway Routing Protocol: BGP

Introduction to socket programming- Concurrent Processing in Client-Server Software-Byte ordering and address conversion functions – Socket Interface - System calls used with sockets - Iterative server and concurrent server- Multi protocol and Multi service server- TCP/UDP Client server programs – Thread Creation and Termination – TCP Echo Server using threads-Remote Procedure Call.

Symmetric ciphers: Classical Encryption Techniques: Substitution Techniques,Transposition Techniques, Steganography. Block Ciphers and the Data Encryption Standard, Block Cipher Principles, The Data Encryption Standard. Basic Concepts in Number Theory and Finite Fields: Divisibility and the Division Algorithm, The Euclidean Algorithm. Advanced Encryption Standard, Pseudorandom Number Generation and Stream Ciphers.

Reading:

1. Richards Stevens, Unix network programming, Vol I & Vol II 4th edition, PHI 2007
2. Stallings , Cryptography and Network Security, Pearson Education 2007

CS5153	Advance Data Mining	PCC	4 – 0 – 0	4 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Analyze Algorithms for sequential patterns.
CO2	Extract patterns from time series data.
CO3	Develop algorithms for Temporal Patterns.
CO4	Identify computing frameworks for Big Data analytics.
CO5	Extend the Graph mining algorithms to Web Mining.

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	1	2	2	1	2		1	
CO2		2	1	1			1	
CO3	1	2	1	1			1	
CO4	2	2	1	1	1	1		
CO5	1	2	2	1	2		1	

Detailed syllabus

Sequential Pattern Mining concepts, primitives, scalable methods; Transactional Patterns and other temporal based frequent patterns, Mining Time series Data, Periodicity Analysis for time related sequence data, Trend analysis, Similarity search in Time-series analysis; Mining Data Streams, Methodologies for stream data processing and stream data systems, Frequent pattern mining in stream data, Sequential Pattern Mining in Data Streams, Classification of dynamic data streams, Class Imbalance Problem; Graph Mining, Mining frequent subgraphs, finding clusters, hub and outliers in large graphs, Graph Partitioning; Web Mining, Mining the web page layout structure, mining web link structure, mining multimedia data on the web, Automatic classification of web documents and web usage mining; Distributed Data Mining, Distribute data mining framework, Distributed data source, Distributed data mining techniques, Distributed classifier learning, distributed clustering, distributed association rule mining and Challenges of distributed data mining; Social Network Analysis, characteristics of social Networks.

Reading:

1. Jiawei Han and M Kamber , Data Mining Concepts and Techniques, , Second Edition, Elsevier Publication, 2011.
2. Vipin Kumar, Introduction to Data Mining - Pang-Ning Tan, Michael Steinbach, Addison Wesley, 2006.
3. G Dong and J Pei, Sequence Data Mining, Springer, 2007.
4. Research Papers

CS5154	High Performance Computing Lab	PCC	0 – 0 – 3	2 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Implement and fine tune programs for matrix operations using OpenMP, Pthreads, MPI CUDA languages/libraries.
CO2	Analyze the working group communication operations of MPI
CO3	Design and implement sorting algorithms using OpenMP, Pthreads, MPI and CUDA language/libraries.
CO4	Design and implement algorithms for elementary problems of Graphs using OpenMP, Pthreads, MPI and CUDA language/libraries

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	1	2	2		1		1	1
CO2		1	2	1	1		1	
CO3	1	2	1	1			1	
CO4	1	2	2	1	1		1	1

Detailed syllabus

1. Matrix-Matrix multiplication – simple, Cannon's, DNS algorithm
2. Gaussian elimination – simple, partial pivoting
3. Back-substitution (triangular system of equations)
4. Soting – Bitonic, Odd-even transposition, Shellsort, Quicksort, Bucket, Radix
5. Minimum spanning tree
6. Single source shortest paths
7. All-pairs shortest paths – Dijkstra's algorithm, Floyd's algorithm
8. Transitive closure
9. Connected components
10. Maximal independent set for sparse graphs

Solve each one of the above problems using OpenMP, Pthreads, MPI, Intel TBB, CUDA, OpenCL

Reading:

1. Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar : Introduction to Parallel Computing, Second Edition Pearson Education, 2007
2. Benedict R Gaster, Lee Howes, David R Kaeli Perhaad Mistry Dana Schaa, *Heterogeneous Computing* with OpenCL McGraw-Hill, Inc. Newyork, 2011
3. Michael J. Quinn , Parallel Programming in C with MPI and OpenMP McGraw-Hill International Editions, Computer Science Series, 2004
4. Jason Sanders, Edward Kandrot, CUDA By Example – An Introduction to General-Purpose GPU Programming, Addison Wesley, 2011

CS5155	Socket Programming and Network Security Lab	PCC	0 – 0 – 3	2 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Develop client-server based applications.
CO2	Analyze security vulnerabilities in TCP/IP.
CO3	Develop secure protocols for different types of attacks.

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	1	2	3	3	2	2	1	1
CO2	2	2	3	2	2	1	1	2
CO3	2	3	3	3	2	2	2	2

Detailed syllabus

Assignment-1: Implementation of group communication using message queues.

Assignment-2: Using PIPEs and Signals, implement the following IPC program: Nodes are connected in a circular fashion. A token is passed between nodes. Token is initialized to some integer value. When the token reaches a node, Node decrements the token value and Passes the token to the next node. When the token value becomes 0, that particular node generates another token with the initial value, Passes the token to the next node and Kills itself. Node give signals to adjacent nodes .Continue this until all nodes are killed.

Assignment-3: Write a program for a modified tic-tac-toe game using semaphore:

Assignment-4: Write a socket program for authentication problem.

Assignment-5: Write a program for implementation of secure group communication.

Assignment-6: Write a program to implement TCP quiz.

Assignment-7: Write a client-server program, using sockets (TCP), where the client sends a line of characters, to the server, and the server reads the line, from its network input and echoes the line back to the client. (Client will display the same line).

Assignment-8: Repeat assignment 7 using UDP sockets.

Assignment-9: Write a program (Client-Server) that uses an unnamed stream pipe, to connect two processes, from two different systems, where one process (Client) copies stdin to stream pipe, and the other process (Server) copies the stream pipe to stdout.

Assignment-10: Write a program which implements the broadcasting technique.

Assignment-11: Write a program which implements the multicasting technique.

Assignment-12: Simulate a traditional routing protocol using NS2/NS3 simulator (or any other network simulator), for 1) A traditional Internet routing protocol and a secure routing protocol. 2)Wireless network (AODV, DSR).

Reading:

1. J. Thomas Shaw, “Information Security Privacy”, ABA, 2012.
2. Matthew Bailey, “Complete Guide to Internet Privacy, Anonymity and Security”, Nerel Online, 2011.
3. D. S. Herrmann, “A complete guide to security and privacy metrics”, Auerbach Publisher (Taylor and Francis Group), 2007
4. A. Abraham, “Computational Social Networks: security and privacy”, Springer, 2012

CS5191	Seminar	PCC	0 – 0 – 3	2 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Analyze the selected topic, organize the content and communicate to audience in an effective manner
CO2	Practice the learning by self study

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	2	2	1	1		3	2	1
CO2	2	1		1		1	2	2

CS6142	Comprehensive Viva	PCC	0 – 0 – 0	4 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	To comprehend and correlate the understanding of all courses in post graduate curriculum of Computer Science and Engineering

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	2	2	2	1	1	2	1	2

ELECTIVE COURSES

CS5111	Machine Learning	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Identify instance based learning algorithms
CO2	Design neural network to solve classification and function approximation problems
CO3	Build optimal classifiers using genetic algorithms
CO4	Analyze probabilistic methods for learning

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	2	2		2	3		2	2
CO2	3	3	2	2	3		2	2
CO3	3	3	2	2	2		2	2
CO4	3	3	2	2	2		1	1

Detailed syllabus

INTRODUCTION – Well defined learning problems, Designing a Learning System, Issues in Machine Learning; - THE CONCEPT LEARNING TASK - General-to-specific ordering of hypotheses, Find-S, List then eliminate algorithm, Candidate elimination algorithm, Inductive bias - DECISION TREE LEARNING - Decision tree learning algorithm-Inductive bias- Issues in Decision tree learning; - ARTIFICIAL NEURAL NETWORKS – Perceptrons,Gradient descent and the Delta rule, Adaline, Multilayer networks, Derivation of backpropagation rule-Backpropagation Algorithm- Convergence, Generalization; – EVALUATING HYPOTHESES – Estimating Hypotheses Accuracy, Basics of sampling Theory, Comparing Learning Algorithms; - BAYESIAN LEARNING – Bayes theorem, Concept learning, Bayes Optimal Classifier, Naïve Bayes classifier, Bayesian belief networks, EM algorithm; - COMPUTATIONAL LEARNING THEORY – Sample Complexity for Finite Hypothesis spaces, Sample Complexity for Infinite Hypothesis spaces, The Mistake Bound Model of Learning; - INSTANCE-BASED LEARNING – k-Nearest Neighbor Learning, Locally Weighted Regression, Radial basis function networks, Case-based learning - GENETIC ALGORITHMS – an illustrative example, Hypothesis space search, Genetic Programming, Models of Evolution and Learning; Learning first order rules-sequential covering algorithms-General to specific beam search-FOIL; REINFORCEMENT LEARNING - The Learning Task, Q Learning.

Reading:

1. Tom.M.Mitchell, Machine Learning, McGraw Hill International Edition
2. Ethern Alpaydin, Introduction to Machine Learning. Eastern Economy Edition, Prentice Hall of India, 2005.

CS5112	Human Computer Interaction	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Design and Development processes and life cycle of Human Computer Interaction
CO2	Analyze product usability evaluations and testing methods.
CO3	Apply the interface design standards/guidelines for cross cultural and disabled users.
CO4	Categorize, Design and Develop Human Computer Interaction in proper architectural structures.

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	2	3		2	2			
CO2	1		3					1
CO3	2	2	1	3		2	2	1
CO4	1	1	1	3	2			1

Detailed syllabus

HCI foundations- Input–output channels, Human memory, Thinking: reasoning and problem solving, Emotion, Individual differences, Psychology and the design of interactive systems, Text entry devices, Positioning, pointing and drawing, Display devices, Devices for virtual reality and 3D interaction, Physical controls, sensors and special devices, Paper: printing and scanning

Designing- Programming Interactive systems- Models of interaction, Frameworks and HCI, Ergonomics, Interaction styles, Elements of the WIMP interface, The context of the interaction, Experience, engagement and fun, Paradigms for interaction,

Cantered design and testing- Interaction design basics-The process of design, User focus, Scenarios, Navigation design, Screen design and layout, Iteration and prototyping, Design for non-Mouse interfaces, HCI in the software process, Iterative design and prototyping, Design rules, Principles to support usability, Standards and Guidelines, Golden rules and heuristics, HCI patterns

Implementation support - Elements of windowing systems, Programming the application, Using toolkits

User interface management systems, Evaluation techniques, Evaluation through expert analysis, Evaluation through user participation, Universal design, User support

Models and Theories - Cognitive models, Goal and task hierarchies, Linguistic models, The challenge of display-based systems, Physical and device models, Cognitive architectures

Collaboration and communication - Face-to-face communication, Conversation, Text-based communication, Group working, Dialog design notations, Diagrammatic notations, Textual dialog notations, Dialog semantics, Dialog analysis and design

Human factors and security - Groupware, Meeting and decision support systems, Shared applications and artifacts, Frameworks for groupware Implementing synchronous groupware, Mixed, Augmented and Virtual Reality

Reading:

1. A Dix, Janet Finlay, G D Abowd, R Beale., Human-Computer Interaction, 3rd Edition, Pearson Publishers,2008.
2. Shneiderman, Plaisant, Cohen and Jacobs, Designing the User Interface: Strategies for Effective Human Computer Interaction, 5th Edition, Pearson Publishers, 2010.

CS5113	Computer Vision	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Classify Image representations
CO2	Apply Image transformation methods
CO3	Implement image processing algorithms
CO4	Design of face detection and recognition algorithms

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	1	1	1					
CO2		1			1			
CO3	2	1		2	1		2	
CO4	2	2		1			2	1

Detailed syllabus

The image model and acquisition, image shape, sampling, intensity images, color images, range images, image capture, scanners. Statistical and spatial operations, Gray level transformations, histogram equalization, multi image operations. Spatially dependent transformations, templates and convolution, window operations, directional smoothing, other smoothing techniques. Segmentation and Edge detection, region operations, Basic edge detection, second order detection, crack edge detection, edge following, gradient operators, compass & Laplace operators. Morphological and other area operations, basic morphological operations, opening and closing operations, area operations, morphological transformations. Image compression: Types and requirements, statistical compression, spatial compression, contour coding, quantizing compression. Representation and Description, Object Recognition, 3-D vision and Geometry, Digital Watermarking. Texture Analysis.

Reading:

1. D. A. Forsyth, J. Ponce , Computer Vision: A Modern Approach, , PHI Learning 2009.
2. Milan Soanka, Vaclav Hlavac and Roger Boyle, Digital Image Processing and Computer Vision- Cengage Learning, 2014
3. R.C. Gonzalez and R.E. Woods, Digital Image Processing, Pearson Education, 2007

CS5114	Intelligent Agents	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Search for solution using A*, Mini-Max algorithms.
CO2	Create logical agents to do inference using first order logic.
CO3	Apply Bayesian Networks for probabilistic reasoning.
CO4	Perform Statistical learning using EM algorithm.

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	2	2		1	1			
CO2	3	2		1				
CO3	2			1				
CO4	1	2						

Detailed syllabus

INTRODUCTION – Agents and Objects – Evaluation of Agents – Agent Design Philosophies - Multi-agent System – Mobile Agents – Agent Communication – Knowledge query and Manipulation Language – Case Study. INTRODUCTION – What is AI? , The Foundations of Artificial Intelligence; - INTELLIGENT AGENTS – Agents and Environments, Good Behavior: The Concept of Rationality, The Nature of Environments, The Structure of Agents; - SOLVING PROBLEMS BY SEARCH – Problem-Solving Agents, Formulating problems, Searching for Solutions, Uninformed Search Strategies, Breadth-first search, Depth-first search, Searching with Partial Information, Informed (Heuristic) Search Strategies, Greedy best-first search, A* Search: Minimizing the total estimated solution cost, Heuristic Functions, Local Search Algorithms and Optimization Problems, Online Search Agents and Unknown Environments; –ADVERSARIAL SEARCH – Games, The minimax algorithm, Optimal decisions in multiplayer games, Alpha-Beta Pruning, Evaluation functions, Cutting off search, Games that Include an Element of Chance; - LOGICAL AGENTS – Knowledge-Based agents, The Wumpus World, Logic, Propositional Logic: A Very Simple Logic, Reasoning Patterns in Propositional Logic, Resolution, Forward and Backward chaining; - FIRST ORDER LOGIC – Syntax and Semantics of First-Order Logic, Using First-Order Logic , Knowledge Engineering in First-Order Logic; - INFERENCE IN FIRST ORDER LOGIC – Propositional vs. First-Order Inference, Unification and Lifting, Forward Chaining, Backward Chaining, Resolution ; - UNCERTAINTY – Acting under Uncertainty, Basic Probability Notation, The Axioms of Probability, Inference Using Full Joint Distributions, Independence, Bayes' Rule and its Use, The Wumpus World Revisited; - PROBABILISTIC REASONING – Representing Knowledge in an Uncertain Domain, The Semantics of Bayesian Networks, Efficient Representation of Conditional Distribution, Exact Inference in Bayesian Networks, Approximate Inference in Bayesian Networks; - STATISTICAL LEARNING METHODS –

Statistical Learning, Learning with Complete Data, Learning with Hidden Variables: EM Algorithm.

Reading:

1. Stuart Russell, Peter Norvig, Artificial Intelligence -A Modern Approach, 2/e, Pearson, 2003.
2. Nils J Nilsson, Artificial Intelligence: A New Synthesis, Morgan Kaufmann Publications, 2000.

CS5121	Object Oriented Software Engineering	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Apply the Object Oriented Software-Development Process to design software
CO2	Analyze and Specify software requirements through a SRS documents.
CO3	Design and Plan software solutions to problems using an object-oriented strategy.
CO4	Model the object oriented software systems using Unified Modeling Language (UML)
CO5	Estimate the cost of constructing object oriented software.

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	1	3	3	3	3	3	2	2
CO2		3	3	2	2	2	1	
CO3	1	3	2	3	3	3	1	1
CO4		3	2	3	2	3	1	1
CO5	1	3	3	3	3	3	2	2

Detailed syllabus

Object oriented Paradigm, Object oriented Concepts, Classes, Objects, Attributes, Methods and services, Messages, Encapsulation, Inheritance, Polymorphism, Identifying the elements of object model, management of object oriented Software projects, Object Oriented Analysis, Domain Analysis, Generic Components of OOA model,, OOA Process, Object Relationship model, Object Behavior Model, Object Oriented Design: Design for Object- Oriented systems, The Generic components of the OO design model, The System design process, The Object design process, Design Patterns, Object Oriented Programming. Object Oriented testing: Broadening the view of Testing, Testing of OOA and OOD models, Object-Oriented testing strategies, Test case design for OO software, testing methods applicable at the class level, Interclass test case design . Technical Metrics for Object Oriented Systems: The Intent of Object Oriented metrics, The distinguishing Characteristics, Metrics for the OO Design model, Class-Oriented metrics, Operation-Oriented Metrics, Metrics foe Object Oriented testing, Metrics for Object Oriented projects. Computer-Aided Software Engineering: What is CASE?, Building blocks for CASE, A taxonomy of CASE tools, Integrated CASE environments, The Integration Architecture, The CASE Repository.

Reading:

1. Stephen R. Schach, Object Oriented and Classical Software Engineering, 5th Edition, TMH, 2010
2. Roger S. Pressman, Software Engineering - A Practitioner's Approach, 6th Edition, MGH, 2005.

CS5122	Wireless and Mobile Networks	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Identify issues related to environment, communication, protocols in mobile computing
CO2	Evaluate the performance of mobile IPv and IPv6 architectures
CO3	Analyze the performance of mac protocols for wired and wireless networks
CO4	Analyze performance of transport layer protocols in mobile Ad-hoc networks
CO5	Design and analyze routing protocols for multi-hop wireless networks

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	1			2	3	1	2	1
CO2		2	3	1	1			1
CO3	1	2	2	2	1		2	
CO4	1	2	2	2	1		1	
CO5	1	2	2	2	1		1	

Detailed syllabus

Basic communication Technologies, Introduction to Mobile Networks, Types of Wireless networks (MANET: Mobile ad-hoc networks, WSN: Wireless Sensor Networks, VANET: Vehicular Ad-hoc Networks, PAN: Personal Area Networks, DTN: Delay Tolerant Network), Wireless Communication Fundamentals, Cellular Wireless Networks, Mobile Ad-hoc Networks, Medium Access Control Layer: MACA, MACAW, Wireless LAN, Mobile Network Layer (Mobile IP), DHCP, Routing in Mobile Ad hoc Networks (MANET): AODV (Ad-hoc On-Demand Distance Vector Routing Protocol), DSR (Dynamic Source Routing), Secure routing protocols in MANET, Wireless Sensor Networks: (Routing protocols, Localization methods, Sensor Deployment Strategies), Delay Tolerant Networks, Vehicular Ad-hoc Networks, Wireless Access Protocol, GPS, RFID.

Reading:

1. C D M Cordeiro, D. P. Agarwal, “Adhoc and Sensor Networks: Theory and applications”, World Scientific, 2006.
2. Jochen Schiller, “Mobile Communications”, Second Edition, Pearson Education, 2003.
3. Asoke K Talukder and Roopa R. Yavagal; Mobile Computing – Technology, Applications and Service Creation; TMH Pub., New Delhi, 2006.

CS5123	Real Time Systems	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Identify multi tasking techniques in real time systems.
CO2	Evaluate the performance of soft and hard real time systems.
CO3	Analyze multi task scheduling algorithms for periodic, aperiodic and sporadic tasks.
CO4	Design real time operating systems.

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	2	2	2	2	2			1
CO2	2	2	2	2	2		2	1
CO3	1	2	2	2	2			
CO4	2	3	2	3	3	2	1	1

Detailed syllabus

Real Time Applications : Digital control systems, High level control systems, Signal processing, Real time databases, Multimedia applications. Hard versus Soft Real Time Systems : Jobs and Processors, Release Times, Deadlines and Timing constraints, Hard and soft timing constraints, hard real time systems and soft real time systems. Reference Model for Real Time Systems : Processors and resourses, temporal parameters of real-time workload, Periodic task model, Precedence constraints and data dependency, Temporal dependency, AND/OR Precedence constraints, conditional branches, pipeline relationship, Functional parameters, Resource parameters of jobs and parameters of resources, Scheduling hierarchy.

Real Time Scheduling Approaches : Clock-driven approach, weighted round robin approach, Priority driven approach, dynamic versus static systems, Effective release and times and deadlines, Optimality of EDF and LST algorithms, Challenges in validating timing constraints in priority driven systems, Offline versus online scheduling. Clock Driven and Priority Driven Scheduling : static, timer-driven scheduler, general structure of cyclic schedules, cyclic executives, improving the average response time of aperiodic jobs, scheduling sporadic jobs, practical considerations and generalizations, pros and cons of clock-driven scheduling, fixed-priority versus dynamic-priority algorithms, maximum schedulable utilization, optimality of the rm and dm algorithms, sufficient schedulability conditions for the rm and dm algorithms, scheduling aperiodic and sporadic jobs in priority driven systems, deferrable servers, sporadic servers, constant utilization, total bandwidth, and weighted fair-queueing servers, scheduling of sporadic jobs, a two-level scheme for integrated scheduling, Resources and Resource Access Control : assumptions on resources and their usage, effects of resource contention and resource access control, nonpreemptive critical sections, basic priority-inheritance protocol, basic priority-ceiling protocol, stack-based, priority-ceiling (ceiling-priority) protocol, use of priority-ceiling protocol in dynamic-priority systems, preemption-ceiling protocol, controlling accesses to multiple-unit resources, controlling concurrent accesses to data objects.Operating Systems : overview, time services and scheduling mechanisms, other basic operating system functions, processor reserves and resource kernel, open system architecture, capabilities of commercial real-time operating systems, predictability of general-purpose operating systems.

Reading:

1. Jane Liu , *Real-Time Systems,* Prentice Hall, 2000.
2. Philip.A.Laplante, *Real Time System Design and Analysis, 3/e, PHI, 2004.*

CS5124	Model Driven Software Development	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Apply software development techniques with reference to model driven software development
CO2	Design and implement the practical application of domain-specific modeling language.
CO3	Identify verification and translation of specifications.
CO4	Analyze emerging model-driven development techniques.
CO5	Integrate a set of models to perform effective software specifications.

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	2	2	2	2	3			1
CO2	1	2	2	1	2	1	2	1
CO3	1	1	2	1	2		1	1
CO4	1	2	1	1	1		1	1
CO5		2	2	2	2		1	1

Detailed syllabus

MDSD basic ideas and terminology: The challenges, The goals of MDSD, MDSD approach, architecture. Case study: a typical web application. Concept formation: Common MDSD concepts and terminology, model driven architecture, architecture centric MDSD, Generative Programming. Classification: MDSD vs CASE, 4GL, wizard, roundtrip engineering, MDSD and Patterns, MDSD and domain driven design. MDSD capable target architecture: Software Architecture in the context of MDSD. Building blocks of software architecture. Architecture reference model, balancing the MDSD platform, MDSD and CBD, SOA, BMP.

Building domain architecture: DSL construction, General transformation architecture, technical aspects of building transformations, the use of interpreters. Code generation techniques: categorization, generation techniques Model transformations with QVT, M2M language requirements. MDSD tools: roles, architecture, selection criterion and pointers.

Software processes - modular-based software design - Model-driven Architecture (MDA): What is metamodelig, Metalevels vs Levels of abstraction, MDA Frameswork: Platform Independent Model PIM and Platform Specific Model. System modeling- MOF's metamodelling

.

Reading:

1. Thomas Stahl, Markus Voelter, Model-Driven Software Development: Technology, Engineering, Management, Wiley, 2006.
2. Anne Kleppe, Jos Warmer and Wim Bast. MDA Explained. The Model Driven Architecture, Practice and Promise, Pearson Education, Boston, USA, 2003.

CS5131	Distributed Computing	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Identify models of distributed computing
CO2	Analyze algorithms for coordination, communication, security and synchronization in distributed systems
CO3	Classify distributed shared memory models
CO4	Design and Implement distributed file systems
CO5	Design distributed algorithms for deadlocks

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	1	2	2	1	2		1	1
CO2	2	2	1	1	1	2	1	1
CO3		2	1	2	1		1	
CO4		2	2	2	2			
CO5	1	2	2		1			

Detailed syllabus

Distributed Computing Introduction : Definition, Relation to parallel systems, synchronous vs asynchronous execution, design issues and challenges A Model of Distributed Computatioms : A Model of distributed executions, Models of communication networks, Global state of distributed system, Models of process communication. Logical Time : Logical clocks, scalar time, vector time, Efficient implementation of vector clocks, Jard-Jourdan's adaptive technique, Matrix time, vrtual time, Physicla clock synchronization : NTP

Global state and snapshot recording algorithms: System model, Snapshot algorithms for FIFO channels, Variations of Chandy-Lamport algorithm, Snapshot algorithms for non-FIFO channels, Snapshots in a causal deluvery system, Monitoring global state, Nececssary and sufficient conditions for consistent global snapshots, Finding consistent global snapshots in a distributed computation. Message ordering and group communication : Mesage ordering paradigms, Asynchronous execution with synchronous communication, Synchronous program order on an asynchronous system, Group communication, Causal order (CO), Total order, A nomenclature for multicast, Propagation trees for multicast, Classification of application-level multicast algorithms, Semantics of fault-tolerant group communication, Distributed multicast algorithms at the network layer.

Termination detection : Introduction, System model of a distributed computation, Termination detection using distributed snapshots, Termination detection by weight throwing, A spanning-tree-based termination detection algorithm, Message-optimal termination detection, Termination detection in a very general distributed computing model, Termination detection in the atomic computation model, Termination detection in a faulty distributed system Distributed mutual exclusion algorithms: Preliminaries, Lamport's algorithm, Ricart–Agrawala algorithm, Singhal's dynamic information-structure algorithm, Lodha and Kshemkalyani's fair mutual exclusion algorithm, Quorum-based mutual exclusion algorithms, Maekawa's algorithm, Agarwal–El Abbadi quorum-based algorithm, Token-based algorithms, Suzuki–Kasami's broadcast algorithm, Raymond's tree-based algorithm,.Deadlock detection in distributed systems: Introduction, System model, Preliminaries, Models of deadlocks, Knapp's classification of distributed deadlock detection algorithms, Mitchell and Merritt's algorithm for the singleresource model, Chandy–Misra–Haas algorithm for the AND model, Chandy–Misra–Haas algorithm for the OR model, Kshemkalyani–Singhal algorithm for the P-out-of-Q model Distribued shared memory : Abstraction and advantages, Memory consistency models, Shared memory mutual exclusion, Wait-freedom, Register hierarchy and wait-free simulations, Wait-free atomic snapshots of shared objects Check pointing and rollback recovery : Introduction, Background and definitions, Issues in failure recovery, Checkpoint-based recovery, Log-based rollback recovery, Koo–Toueg coordinated checkpointing algorithm, Juang–Venkatesan algorithm for asynchronous checkpointing and recovery, Manivannan–Singhal quasi-synchronous checkpointing algorithm, Peterson–Kearns algorithm based on vector time, Helary–Mostefaoui–Netzer–Raynal communication-induced protocolConsensus and agrrement algorithms : Problem definition, Overview of results, Agreement in a failure-free system (synchronous or asynchronous), Agreement in (message-passing) synchronous systems with failures, Agreement in asynchronous message-passing systems with failures, Wait-free shared memory consensus in asynchronous systems Failure detectors : Introduction, Unreliable failure detectors, The consensus problem, Atomic broadcast, A solution to atomic broadcast, The weakest failure detectors to solve fundamental agreement problems, An implementation of a failure detector, An adaptive failure detection protocol.

Reading:

1. Ajay D. Kshemakalyani, Mukesh Singhal, *Distributed Computing,*Cambridge University Press, 2008.
2. Andrew S. Tanenbaum, Maarten Van Steen, *Distributed Systems - Principles and Paradigms,* PHI, 2004.

CS5132	Advanced Databases	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Design distributed database for application development.
CO2	Apply query optimization principles for optimizing query performance in centralized and distributed database systems
CO3	Design distributed database schema using principles of fragmentation and allocation.
CO4	Apply distributed transaction principles for handling transactions in distributed database applications.
CO5	Apply distributed database administration principles for managing distributed database.

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	2	2	2	2	2			1
CO2	2	1	1	1	2		2	
CO3	1	2	2	1	2		2	1
CO4	1	2	1	2	1			
CO5		1	1	2	1			

Detailed syllabus

Distributed Databases: Introduction to Distributed Database Systems, Distributed Database System Architecture; Top-Down Approach, Distributed Database Design Issues, Fragmentation, Allocation, Database Integration, Bottom-up approach, Schema Matching, Schema Integration, Schema Mapping; Data and Access Control, View Management, Data Security; Query processing problem, Objectives of Query processing,Compluexity of Relational Algebra Operations, Characterization of Query Processors, Layers of Query Processing; Query Decomposition, Normalization, Analysis, Elimination of Redundance and Rewriting; Localization of Distributed Data, Reduction for primary Horizontal, Vertical, derived Fragmentation; Distributed Query Execution, Query Optimization, Join Ordering, Static& Dynamic Approach, Semi-joins, Hybrid Approach;Taxonomy of Concurrency control Mechanisms, Lock-Based Concurrency Control, Timestamp-Based Concurrency Control, Optimistic Concurrency Control, Deadlock Management; Heterogeneity issues Advanced Transaction Models, Distributed systems 2PC& 3PC protocols, Replication protocols, Replication and Failures, Hot Spares;

Parallel Databases: Introduction to Parallel Databases, Parallel Database System Architectures, Parallel Data Placement, Full Partitioning; Parallel Query Processing, Query

Parallelism; Parallel Query Optimization, Search Space, Cost Model, Search Strategy; Load Balancing.

Reading:

1. M T Ozsu, Patrick Valduriez, Principles of Distributed Database Systems, Prentice Hall, 1999.
2. S. Ceri and G. Pelaggati, Distributed Database System Principles and Systems, MGH, 1985.

CS5133	Game Theory	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Analyze games based on complete and incomplete information about the players
CO2	Analyze games where players cooperate
CO3	Compute Nash equilibrium
CO4	Apply game theory to model network traffic
CO5	Analyze auctions using game theory

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	1	1	1	1	1		1	
CO2	2	1	1	1			1	
CO3	2						1	
CO4	2	2	1	1			1	
CO5	2	1		1	1		1	1

Detailed syllabus

Games, Old and New; Games, Strategies, Costs, and Payoffs; Basic Solution Concepts Finding Equilibria and Learning in Games; Refinement of Nash: Games with turns and Subgame Perfect Equilibrium; Nash Equilibrium without Full Information: Bayesian Games; Cooperative Games, Markets and Their Algorithmic Issues; Is the NASH-Equilibrium Problem NP-Copmplete?; The Lemke-Howson Algorithm; The Class PPAD. Succinct Representations of Games; The Reduction; Correlated Equilibria; Bitmatrix Games and Best Response Condition; Equilibria Via Labeled Polytopes; The Lemke-Howson Algorithm; Integer Pivoting and Degenerate Games; Extensive Games and Their Strategic Form; Subgame Perfect Equilibria; Computing Equilibria with Sequence Form.

Model and Priliminaries; External Regret Minimization; Regret minimization and Game Theory; Generic Reduction from External to Swap Regret; On the Convergence of Regret-Minimizing Strategies to Nash Equilibrium in Routing Games; Fisher's Linear Case and the Eisenberg –Gale Convex Program; Checking if Given Prices are Equilibrium Prices; Two Crucial Ingredients of the Algorithm; The Primal-Dual Schema in the Enhanced Setting; Tight Sets and the Invariants; Balanced Flows; The Main Algorithm and Running Time; The Linear-Case of Arrow-Debreu Model; Algorithm for Single-Source Multiple-Sink Markets; Fisher Model with Homogeneous Consumers; Exchange Economics Satisfying WGS; Specific Utility Functions; Computing Nash Equilibria in Tree Graphicial Games; Graphicial Games and Correlated Equilibria; Graphicial Exchange Economies.

Reading:

1. Noam Nisan, Tim Roughgarden, Eva Tardos, Vijay V. Vazirani, *Algorithmic Game Theory*, Cambridge University Press, 2007.
2. Ronald Cohn Jesse Russell, *Algorithmic Game Theory*, VSD Publishers, 2012.

CS5134	Semantic Web	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Analyze the Semantic Web architectures.
CO2	Perform Ontology reasoning.
CO3	Apply Ontology programming using Jena-API.
CO4	Develop Ontology using Protege Editor.
CO5	Perform queries on Ontology.

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	2	2	2	1	2		1	
CO2	2	1	1	1	1			
CO3				2			2	
CO4		1	2	1				
CO5		1	1					1

Detailed syllabus

The Semantic Web Vision, overview of techniques and standards, Semantic Web Architecture, XML with Document Type Definitions and Schemas, Transformation/Inference rules in XSLT, RuleML and RIF, metadata with RDF (Resource Description Framework); metadata taxonomies with RDF Schema; Ontology languages, Ontology Development using Protege editor, Ontology Querying, Ontology Reasoning and Description Logic (DL), Semantic Web Application Areas, Ontology programming with Jena API, Ontology Engineering**.**

Reading:

1. Grigoris Antoniou and Frank van Harmelen, A Semantic Web Primer, 1st Edition, MIT Press, 2004.
2. John Hebeler, Matthew Fisher, Ryan Blace and Andrew Perez-Lopez, Semantic Web Programming, 1st Edition, Wiley, 2009.

CS5135	Big Data Analytics	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Analyze big data challenges in different domains including social media, transportation, finance and medicine
CO2	Explore relational model, SQL and capabilities of emergent systems in terms of scalability and performance
CO3	Apply machine learning algorithms for data analytics
CO4	Analyze the capability of No-SQL systems
CO5	Build secure big data systems
CO6	Analyze MAP-REDUCE programming model for better optimization

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	2	2	3	2	2	2
CO2	3	2	2	2	3	2	2	2
CO3	3	2	2	2	2	1	1	2
CO4	1	2	2	2	2		1	2
CO5	2	2	3	2	2		2	2
CO6	2	2	1	1	2	1		2

Detailed syllabus

Overview of Big Data, Stages of analytical evolution, State of the Practice in Analytics, The Data Scientist, Big Data Analytics in Industry Verticals, Data Analytics Lifecycle, Operationalizing Basic Data Analytic Methods Using R, Advanced Analytics - Analytics for Unstructured Data - Map Reduce and Hadoop, The Hadoop Ecosystem, In-database Analytics, Data Visualization Techniques, Stream Computing Challenges, Systems architecture, Main memory data management techniques, energy-efficient data processing, Benchmarking, Security and Privacy, Failover and reliability.

Reading:

1. Bill Franks, Taming The Big Data Tidal Wave, 1st Edition, Wiley, 2012.
2. Frank J. Ohlhorst, Big Data Analytics,1st Edition, Wiley, 2012.

CS5161	Security and Privacy	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Evaluate the risks and vulnerabilities in protocols/Standards.
CO2	Apply Number Theory and Algebra required for designing cryptographic algorithms.
CO3	Design symmetric key and asymmetric key encryption techniques.
CO4	Design authentication, message integrity and authenticated encryption protocols
CO5	Design and security analysis of systems including distributed storage and Electronic voting.

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	1	2	3	1				1
CO2	1	2	2	1				1
CO3	1	2	2	1				1
CO4	1	2	2	1				1
CO5	1	2	2	1				1

Detailed syllabus

Introduction to Security – risks, threats and vulnerabilities, Cryptography, Stream Ciphers – One-time Pad (OTP), Perfect secrecy, Pesudo-random generators (PRG), Attacks on stream ciphers and OTP, Real world stream ciphers, Semantic security, Case Study- RC4, Salsa 20, CSS in DVD encryption, A5 in GSM, Block ciphers- DES, attacks, AES, Block ciphers from PRG, Modes of operation – one-time key and many-time keys, CBC, CTR modes, Message Integrity – MAC, MAC based on PRF, NMAC, PMAC, Collision resistance – Birthday attack, Merkle-Damgard construction, HMC, Case study:SHA-256, Authenticated encryption, Key exchange algorithms, Public key cryptosystems – RSA, ElGamal, Elliptic curve cryptosystems – PKC, key exchange, IBE, Case studies – HTTPS – SSL/TLS, SSH, IPSec, 802.11i WPA, System design and analysis – Survivable distributed storage system, Electronic voting system.

Reading:

1. J. Thomas Shaw, "*Information Security Privacy*", ABA, 2012.
2. J. Katz and Y. Lindell, Introduction to Modern Cryptography, CRC press, 2008.
3. Menezes, et.al, Handbook of Applied Cryptography, CRC Press, 2004.
4. A. Abraham, "Computational Social Networks: security and privacy", Springer, 2012.

CS5162	Intrusion Detection Systems	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Identify Intrusion in a network
CO2	Classify attack types that affect security of the network.
CO3	Analyze IP network traffic for malicious packet identification.
CO4	Apply data mining techniques for database intrusion detection
CO5	Design database intrusion system for database protection.
CO6	Identify packet sniffing in a network

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	1	2	2		1			
CO2		2	2	1	1		1	
CO3		2	2	1			1	
CO4	1	2	1		1		1	
CO5	1	2	2	1	1		1	
CO6		1	2				1	

Detailed syllabus

Introduction, Some Definitions, Where IDS Should be Placed in Network Topology, Honey Pots, Security Zones and Levels of Trust, IDS Policy, Components of an intrusion detection system - Packet Decoders, Preprocessors, The Detection Engine, Logging and Alerting System, Output Modules, Dealing with Switches, TCP Stream Follow Up, How to Protect IDS Itself, Internal and external threats to data, attacks, need and types of IDS, Intrusion Prevention Systems, Network IDSs, Protocol based IDs, Application protocol, host based, hybrid based IDs, architectures

Snort - Snort Installation Scenarios, Test Installation, Single Sensor Production IDS, Single Sensor with Network Management System Integration, Single Sensor with Database and Web Interface, Multiple Snort Sensors with Centralized Database, Installing Snort, Running Snort on a Non-Default Interface, Automatic Startup and Shutdown, Running Snort on Multiple Network Interfaces, Location of Snort Files, Snort Modes, Snort Alert Modes, Running Snort in Stealth Mode

Working with Snort Rules - TCP/IP Network Layers, CIDR, Structure of a Rule, Rule Headers, Rule Options, The Snort Configuration File, Order of Rules Based upon Action, Automatically

Updating Snort Rules, Default Snort Rules and Classes, Sample Default Rules, Writing Good Rules

Plugins, Preprocessors and Output Modules - Preprocessors - HTTP Decode, Port Scanning, The frag2 Module, The stream4 Module, The spade Module, ARP Spoofing, Output Modules - The alert_syslog Output Module, The alert_full Output Module, The alert_fast Output Module, The alert_smb Module, The log_tcpdump Output Module, The XML Output Module, Logging to Databases, CSV Output Module, Unified Logging Output Module, SNMP Traps Output Module, Log Null Output Module, Using BPF Filters

Using Snort with MySQL - Making Snort Work with MySQL, Snort Compilations with MySQL Support, Install MySQL, Creating Snort Database in MySQL, Creating MySQL User and Granting Permissions to User and Setting Password, Creating Tables in the Snort Database, Modify snort.conf Configuration File, Starting Snort with Database Support, Logging to Database, Secure Logging to Remote Databases Securely, Using Stunnel, Snort Database Maintenance

Using ACID and SnortSnarf with Snort - Installation and Configuration, Using ACID - ACID Main Page, Listing Protocol Data, Alert Details, Searching, Searching whois Databases, Generating Graphs, Archiving Snort Data, ACID Tables, SnortSnarf, Barnyard

Data Mining Techniques, Data correlation, Data fusion Techniques, AI techniques, Artificial Immunity, Agent development for intrusion detection, Architecture models of IDs and IPs, future needs.

SECURITY AND PRIVACY

Introduction: The need of Security and Privacy, Basic concepts: number theory, Formal analysis and design of algorithms and protocols, Provable Security, Cryptosystems; Privacy: Foundations of Privacy, Differential Privacy: Definitions and Early Uses, Privacy Regulations, Noiseless Differential Privacy, Privacy preserving Data Mining techniques.

Privacy & Security and Meaningful Use (MU) - Core Measure, Comply with Security and Privacy Guidelines, Identify Risks to the data, Security Risk Analysis, Mitigate Risks, Risk Management and Security Components, The Threat of Cyber Attacks

Plan for Meeting Privacy and Security Portions of Meaningful Use - Confirm You Are a “Covered Entity”, Provide Leadership, Document Your Process, Findings and Actions, Conduct Security Risk Analysis, Develop an Action Plan, Manage and Mitigate Risks, Information Security Settings, Written Policies and Procedures, Continuous Monitoring of Security Infrastructure, Prevent with Education and Training, Workforce Education and Training, Communicate with stakeholders, Update Business Associate Agreements, Attest for the Security Risk Analysis MU Objective

Integrating Privacy and Security into Practice, Understanding Rights of Owner of the data and Provider Responsibilities, Privacy Rules, Limits on Using & Disclosing Information, How to Keep the Information Secure, Cybersecurity, What to Do in Case of a Breach of Unsecured Information, Comply with Security Rules, Compliance with Other Laws and Requirements

Privacy and Security Resources, Technical Assistance, Regulatory & Guidance Information, Meaningful Use – Privacy & Security, Tools, Education & Training Materials, Brochures, Fact Sheets

Reading:

1. Rafeeq Rehman, Intrusion Detection with SNORT, Apache, MySQL, PHP and ACID, Prentice Hall PTR, 2003.
2. Rebecca Gurley Bace, Intrusion Detection, Technology Series MTP 2007.

CS5163	Social Media Analysis	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Classify social networks
CO2	Analyze social media and networking data
CO3	Apply Social networks Visualization tools
CO4	Analyze the social data using graph theoretic computing approach
CO5	Identify application driven virtual communities from social networks
CO6	Apply sentiment mining

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1		2	1	1			1	
CO2		2	2		2		1	
CO3		1		1		1	2	1
CO4	2	1				1	1	
CO5	1					1	1	1
CO6	2	2			2	2	2	2

Detailed syllabus

Introduction to social network analysis, Vertex or node, edge, neighbors, degree, shortest path, cycle, tree, complete graph, bipartite graphs, directed graphs, weighted graphs, adjacency matrix, connected components, Games on networks, game theory strategies, dominant strategies, dominated strategies, pure strategies and mixed strategies, Nash equilibrium, multiple equilibria-coordination games, multiple equilibria-the Hawk-Dove game, mixed strategies, Modeling network traffic using game theory.Technological networks (internet, telephone network, power grids, transportation networks), social networks (facebook, movie collaboration, paper collaboration), information networks (web), biological networks (neural networks, ecological networks), Random models of networks, Erdos-Renyi model of random graph, models of the small world, decentralized search in small-world , random graphs with general degree distributions, models of network formation, Spread of influence through a network, influence maximization in networks, spread of disease on networks, Information networks, structure of the web, link analysis and web search, page rank, spectral analysis of page rank and hubs and authorities, random walks, auctions and matching markets, sponsored search markets

Reading:

1. David Easley and Jon Kleinberg, *Networks, Crowds, and Markets: Reasoning About a Highly Connected World.,* Cambridge University Press, 2010.
2. Mark Newman, *Networks: An Introduction.,* Oxford University Press, 2010.
3. Hansen, Derek, Ben Sheiderman, Marc Smith., Analyzing Social Media Networks with NodeXL: Insights from a Connected World, Morgan Kaufmann, 2011.
4. Avinash Kaushik., Web Analytics 2.0: The Art of Online Accounta-bility, Sybex, 2009.

CS5164	Randomized and Approximation Algorithms	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Design and analyze efficient randomized algorithms
CO2	Apply tail inequalities to bound error-probability
CO3	Analyze randomized algorithms with respect to probability of error and expected running time.
CO4	Analyze approximation algorithms and determine approximation factor.

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	1	1	1	1	1	1	1
CO2	2							
CO3	2		1	1	2			
CO4	2		1	1	1			1

Detailed syllabus

Introduction, Las Vegas and Monte Carlo Algorithms, Computational Model and Complexity Classes, Game Tree Evaluation, The Markov and Chebyshev Inequalities, The Stable Marriage Problem, The Coupon Collectors Problem, The Chernoff Bound, Routing in a Parallel Computer, The Probabilistic Method: Overview, probablistic analysis, use of indicator random variables, Randomly permuting arrays, Birthday paradox, analysis using indicator random variables, Balls and bins, Streaks, Online hiring problem, Maximum Satisfiability, Expanding Graphs, The Lovasz Local Lemma, Markov Chains, Random Walks on Graphs, Graph Connectivity, Expanders and Rapidly Mixing Random Walks, Pattern Matching, Random Treaps, Skip Lists, Hash Tables, Linear Programming, The Min-Cut Problem, Minimum Spanning Trees, The DNF Counting Problem, Maximal Independent Sets, Perfect Matchings, The Online approximations paging Problem, Adversary Models and Paging against an Oblivious Adversary, Vertex cover problem, traveling salesman problem with triangle inequality, general traveling salesman problem, set-covering problem, a greedy approximation algorithm, analysis Randomization and linear programming, randomized approximation, subset-sum problem, Absolute approximations, Planar Graph Coloring, Maximum Programs Stored Problem, NP-hard Absolute Approximations ε- approximations, Polynomial time approximations schemes, Scheduling Independent Tasks , 0/1 Knapsack, Fully Polynomial time approximations scheme , Rounding , Interval Partitioning , Separation, probalistically good algorithms,

.

Reading:

1. Rajeev Motwani and Prabhakar Raghavan, Randomized Algorithms, Cambridge University Press, 1995.
2. J. Hromkovic, Design and Analysis of Randomized Algorithms, Springer 2005.

CS5171	Pattern Recognition	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Define various classifiers
CO2	Apply Markov Chain and Hidden Markov Models
CO3	Classify the data objects to recognize the patterns based on template matching
CO4	Apply unsupervised learning algorithms to data objects.
CO5	Classify various Clustering algorithms.

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	1							
CO2	2	2	1	2	1		1	
CO3		1		1	1			
CO4	1	2		2	1		1	
CO5	2	2		1	1			1

Detailed syllabus

Classifiers Based on Bayes Decision Theory: Introduction, Bayes Decision Theory, Discriminant Functions and Decision Surfaces, Bayesian classification for Normal Distributions, The Gaussian Probability density function, The Bayesian classifier for normally distributed classes, Estimation of Unknown probability Density functions, Maximum likelihood parameter estimation, Maximum *a Posteriori* Probability estimation, Bayesian Inference, Maximum entropy estimation, Mixture models, Nonparametric estimation, The Naïve-Bayes Classifier. Linear Classifiers: introduction, Linear discriminant functions and Decision hyperplanes, The Perceptron algorithm, Least square methods, Mean square error estimation, Stochastic approximation and the LMS algorithm, Sum of error squares estimation, Mean square estimation revisited, Mean square error regression, MSE estimates posterior class probabilities, The Bias-Variance dilemma, Logistic discrimination, Support Vector machines. Feature Selection: Introduction, Preprocessing, Outlier removal, Data normalization, Missing data, The Peaking Phenomenon, Feature selection based on Statistical hypothesis testing, Hypothesis Testing basics, Application of the t-Test in Feature selection, The Receiver Operating Characteristics (ROC) Curve, Class Seperability Measures, Divergence, Chernoff Bound and Bhattacharyya distance, Scatter matrices, Feature subset selection. Feature Generation: Basis vectors and Images, Singular Value Decomposition, Independent component analysis, Nonlinear Dimensionality Reduction, Discrete Fourier transform, Template Matching: Measures based on Optimal Path Search, Measures based on correlations, Deformable Template Models, Context based Information Retrieval, Markov Chain and Hidden Markov Model, System Evaluation, Unsupervised Learning and Clustering.

Reading:

1. S Theodoridis and K Koutroumbas – Pattern Recognition, 4th Edition, Academic Press, 2009.
2. C Bishop – Pattern Recognition and Machine Learning – Springer, 2006

CS5172	Cloud Computing	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Identify cloud services for application
CO2	Analyze Cloud infrastructure including Google Cloud and Amazon Cloud.
CO3	Analyze authentication, confidentiality and privacy issues in Cloud computing environment.
CO4	Analyze the financial and technological implications for selecting cloud computing platforms.

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	1	2						1
CO2	2		2	2		1		
CO3			2		2			2
CO4	1		2				1	1

Detailed syllabus

Introduction - Cloud Computing Architecture, Cloud Delivery Models, The SPI Framework, SPI Evolution, The SPI Framework vs. the Traditional IT Model, Cloud Software as a Service (SaaS), Cloud Platform as a Service (PaaS), Cloud Infrastructure as a Service (IaaS) Google Cloud Infrastructure - Google File System – Search engine – MapReduce - Amazon Web Services - REST APIs - SOAP API - Defining Service Oriented Architecture, Combining the cloud and SOA, Characterizing SOA, Loosening Up on Coupling, Making SOA Happen, Catching the Enterprise Service Bus, Telling your registry from your repository, Cataloging services, Understanding Services in the Cloud.

Serving the Business with SOA and Cloud Computing, Query API - User Authentication- Connecting to the Cloud - OpenSSH Keys - Tunneling / Port Forwarding - Simple Storage Service - S3, EC2 - EC2 Compute Units, Platforms and storage, EC2 pricing, EC2 customers Amazon Elastic Block Storage - EBS - Ubuntu in the Cloud - Apache Instances in EC2 – Amazon Cloud Services- Amazon Elastic Compute Cloud (Amazon EC2), Amazon SimpleDB, Amazon Simple Storage Service (Amazon S3), Amazon CloudFront, Amazon Simple Queue Service (Amazon SQS), Amazon Elastic MapReduce, Amazon Relational Database Service (Amazon RDS) , EC2 Applications - Web application design - AWS EC2 Capacity Planning – Apache Servers - Mysql Servers - Amazon Cloud Watch - Monitoring Tools.

Reading:

1. Judith Hurwitz, R Bloor, M Kanfman, F Halper, Cloud Computing for Dummies, 1st Edition, Wiley Publishers, 2009.
2. Gautam Shroff, Enterprise Cloud Computing, Cambridge, 2010.
3. Ronald Krutz and Russell Dean Vines, Cloud Security, 1st Edition, Wiley, 2010.

CS5173	Bio-Informatics	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Classify models used in bioinformatics.
CO2	Compute homologues, analyze sequences, construct and interpret evolutionary trees.
CO3	Analyze protein sequences to retrieve protein structures from databases.
CO4	Design of biological data model
CO5	Apply homology modeling and computational drug design.

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	1	2					2	
CO2		1			1		2	
CO3		2	2		1		2	
CO4	1	2					1	
CO5	1	2			2			

Detailed syllabus

Introduction and Biological databases - Introduction, Sequence Alignment - Pair wise sequence alignment, Database similarity searching, Multiple sequence alignment, Profiles and hidden markov models, Molecular Phylogenetics - Phylogenetics basics, Phylogenetic Tree Construction Methods and Programs, Genomics and Proteomics - Genome mapping, assembly and comparison, Functional genomics, Proteomics, Structural Bioinformatics - Basics of protein structure, Protein structure prediction.

Reading:

1. Jin Xiong, Essential Bioinformatics, 1th Edition, Cambridge University Press, 2011.
2. Arthur M Lesk, Introduction to Bioinformatics, 2nd Edition, Oxford University Press, 2007.

CS5174	Design Patterns	DEC	3 – 0 – 0	3 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Identify common design patterns in the context of incremental and iterative development
CO2	Evaluate and refactor software source code using patterns
CO3	Apply design patterns for software design
CO4	Implement design patterns in an object oriented language
CO5	Distinguish pattern approach for software applications

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	1	2	2	2	2		2	1
CO2		2	2	2	2		2	1
CO3	1	2	2	3	2		1	1
CO4		2	2	2	2		1	1
CO5	2	2	2	2	2		2	1

Detailed syllabus

Introduction: What Is a Design Pattern, Design Patterns in Smalltalk MVC, Describing Design Patterns, the Catalog of Design Patterns, Organizing the Catalog, How Design Patterns Solve Design Problems, How to Select a Design Pattern, How to Use a Design Pattern.

A Case Study: Designing a Document Editor: Design Problems, Document Structure, Formatting, Embellishing the User Interface, and Supporting Multiple Look-and-Feel Standards, Supporting Multiple Window Systems, User Operations, Spelling Checking and Hyphenation.

Creational Patterns: Abstract Factory, Builder, Factory Method, Prototype, Singleton.

Structural Pattern: Adapter, Bridge, Composite, Decorator, Façade, Flyweight, Proxy.

Behavioral Patterns: Chain of Responsibility, Command, Interpreter, Iterator, Mediator, Memento, Observer, State, Strategy, Template Method, Visitor, a Brief History, and the Pattern Community

Reading:

1. Erich Gamma, Design Patterns, Addison-Wesley, 1994.
2. Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, Michael Stal, Pattern-Oriented Software Architecture: A System of Pattern, John Wiley & Sons; 1996.

CS6149	Dissertation Part-A	PRC	0 – 0 – 0	8 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Identify the problem of a research project through literature survey.
CO2	Analyze the technical feasibility of the project.
CO3	Propose the solution for the research problem.
CO4	Analyze and design the proposed solution using software engineering practices.

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	2	2	2	2	2	2	3	2
CO2	2	2	2	2	1	2	1	1
CO3	2	2	1	1	1	2	2	2
CO4	2	3	2	1	2	1	2	

CS6199	Dissertation Part-B	PRC	0– 0 – 0	18 Credits

Pre-requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO1	Synthesize and apply prior knowledge to designing and implementing solutions to open-ended computational problems while considering multiple realistic constraints
CO2	Design and Develop the software with software engineering practices and standards
CO3	Analyze Database, Network and Application Design methods
CO4	Evaluate the various validation and verification methods
CO5	Practice CASE tools for solving software engineering CASE Studies
CO6	Analyze professional issues, including ethical, legal and security issues, related to computing projects

Mapping of course outcomes with program outcomes

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	1	2	1	2	2	1
CO2	2	2	1	2		2	1	1
CO3	2	2			1	2		
CO4		2	1			1		1
CO5		2		3				1
CO6		2	2	2				1